

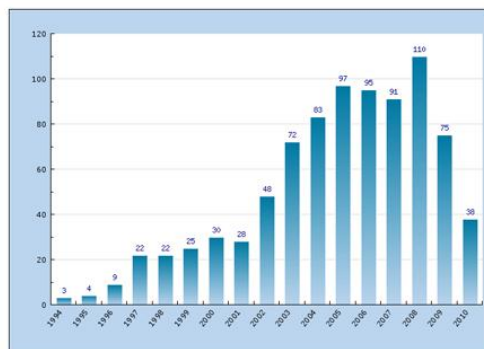
Bachelorarbeit 2010

Studiengang Wirtschaftsinformatik

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen



MARCXML



Student : Samuel Werner

Dozent : Jean Hennebert

Inhaltsverzeichnis

Abbildungsverzeichnis.....	iii
Abstract.....	iv
1 Einleitung	1
1.1 Hintergrund	1
1.2 Aufgabenstellung	1
2 Beschreibung des Ist-Zustands	3
2.1 Aigaion.....	3
2.2 CodeIgniter.....	4
2.3 Datenbank	6
3 Analyse und Auswahl der Möglichkeiten.....	7
3.1 Export von Daten	7
Export nach Microsoft Excel	7
Export für RERO	8
3.2 Generierung von Statistiken.....	8
Flot.....	8
GraPHPite.....	9
Jpgraph.....	9
3.3 Erhöhung der Sichtbarkeit.....	9
URL Rewriting	9
Mailing Lists	10
4 Implementierung der Erweiterungen.....	11
4.1 Verwendete Software	11
4.2 Modifikationen am Linux Server	12
4.3 Export von Daten	12
CSV Export.....	12
MARCXML Export.....	15
4.4 Generierung von Statistiken mit jpgraph	18
4.5 Erhöhung der Sichtbarkeit.....	24
Mailing Lists	26
4.6 Übersetzung.....	29
5 Schlussfolgerungen.....	30

5.1 Fazit.....	30
5.2 Ausblick und Erweiterbarkeit	30
Quellenverzeichnis	32
Glossar.....	33
Bestätigung.....	35
Anhang	36
A Planung	
B Wochenrapporte	
C Pflichtenheft	

Abbildungsverzeichnis

Abbildung 1: Webseite http://publications.hevs.ch	4
Abbildung 2: Funktionsweise CodeIgniter	5
Abbildung 3: Datenbanktabellen	6
Abbildung 4: HTML-Tabelle in Excel dargestellt.....	7
Abbildung 5: HTML-Tabelle in Excel dargestellt.....	8
Abbildung 6: Export vor Erweiterung	12
Abbildung 7: Auszug „DATETITLEAUTHORTYPE.xml“	13
Abbildung 8: Export eines Autors	14
Abbildung 9: Vereinfachtes Schema CSV-Export	14
Abbildung 10: Export nach CSV-Erweiterung	15
Abbildung 11: Auszug MARCXML Struktur RERO.....	15
Abbildung 12: Funktion zur Spracheerkennung	16
Abbildung 13: Info-Popup MARC21	17
Abbildung 14: Export nach CSV- und MARCXML-Erweiterung	17
Abbildung 15: Vereinfachtes Schema MARCXML-Export	18
Abbildung 16: Funktion zur Erstellung eines Balkendiagramms	19
Abbildung 17: Menüpunkt Statistiken	20
Abbildung 18: Auswahlformular zur Anzeige eines Graphs.....	20
Abbildung 19: Füllen der X- und Y-Achse mit Daten.....	21
Abbildung 20: Erstellte Balkendiagramme.....	22
Abbildung 21: Diagrammtypen	23
Abbildung 22: Vereinfachtes Schema Grapherstellung.....	24
Abbildung 23: CI Routing	25
Abbildung 24: Schema URL Rewriting.....	26
Abbildung 25: View show_publication.php	27
Abbildung 26: Meldungen nach E-Mail Versand.....	27
Abbildung 27 Vereinfachtes Schema Mailing List.....	28

Abstract

Die vorliegende Bachelorarbeit behandelt die Analyse und Implementierung von Erweiterungen der Seite www.publications.hevs.ch, welche der HES-SO Sierre als Verwaltungssystem für wissenschaftliche Publikationen dient.

In einem ersten Schritt werden die Struktur und der Code der Webseite studiert, welche Grundlage für die Erweiterungen ist. Danach wird mit Hilfe von Internetrecherchen eine Analyse der Erweiterungsmöglichkeiten durchgeführt und jeweils die beste Lösung ausgewählt und implementiert.

Zu den Erweiterungen gehört erstens der Export der Metadaten von Publikationen nach Excel, damit die Informationen im jährlichen Bericht der HES-SO tabellarisch dargestellt werden können, sowie der Export nach MARCXML, so dass der Westschweizer Bibliotheksverbund (RERO) die Publikationen in ihre Bibliothek aufnehmen kann. Der Export ist ausserdem mit einem Filter versehen, mit welchem die Publikationen nach verschiedenen Kriterien sortiert und nach Jahreszahl exportiert werden können.

Die zweite Erweiterung ist eine Generierung von Grafiken, welche die Anzahl jährlichen Publikationen (nicht kumuliert und kumuliert) darstellt. Hierbei kann nach Institut/Thema, Autor, Zeitspanne, Publikationstyp gefiltert werden. Ausserdem können verschiedene Diagrammtypen ausgewählt werden.

Mit der letzten Erweiterung wird die Sichtbarkeit der Seite erhöht, damit sie von Suchmaschinen besser gefunden wird. Dies wird mittels Vereinfachen und Optimieren der URLs realisiert. Ausserdem wird eine E-Mail an alle Mitglieder versendet, die ein Thema zu welchem gerade eine Publikation hinzugefügt wird, abonniert haben.

1 Einleitung

1.1 Hintergrund

Die HES-SO Sierre hat kürzlich ein Verwaltungssystem für wissenschaftliche Publikationen realisiert. Dieses befindet sich unter <http://publications.hevs.ch>. Die Webseite basiert dabei auf dem Open Source Tool Aigaion, welches öffentlich verfügbar ist und weltweit verwendet wird [1]. Das aktuelle System ist sehr leistungsstark für die grundlegende Verwaltung von erstellten Publikationen. Dabei ist es möglich, die Bibliografien der Publikationen sehr einfach in das System aufzunehmen, zu verändern oder zu löschen. Im Hintergrund läuft eine MySQL-Datenbank.

Einige Funktionalitäten sind jedoch stark limitiert. In der Arbeit geht es darum, vordefinierte Erweiterungen in die bestehende Webseite zu implementieren.

1.2 Aufgabenstellung

Die Erweiterungen können in drei Funktionalitäten gegliedert werden. Dabei sollen zuerst jeweils verschiedene Möglichkeiten analysiert und die optimalste ausgewählt und implementiert werden.

Als erstes soll ein Export der Bibliografien nach Excel realisiert werden. Die generierte Excel-Datei soll primär als Hilfsmittel für das Institut für Wirtschaftsinformatik HES-SO dienen, welches diese für ihren jährlichen Rapport benutzen will. Ausserdem sollen zusätzlich Filter eingebaut werden, mit denen die Publikationen sortiert und nach Jahreszahl exportiert werden können. Damit der Westschweizer Bibliotheksverbund RERO die Publikationen des Instituts importieren kann, sollte des Weiteren ein Export ins MARCXML-Format realisiert werden.

Die zweite Funktionalität umfasst das Generieren von Statistiken, welche die Anzahl Publikationen pro Jahr anzeigen.

Mit der dritten Funktionalität soll die Sichtbarkeit erhöht werden. Das soll mittels URL Rewriting geschehen, damit die Seite besser von Suchmaschinen gefunden wird. Ausserdem sollen Mailing Lists erstellt werden, damit entsprechende Mitglieder automatisch eine E-Mail erhalten, sobald eine neue Publikation veröffentlicht wird.

Darüber hinaus wurden noch folgende optionale Erweiterungen definiert:

- Aufbereitung und Veröffentlichung der Lösungen im Sinne einer Erweiterung von Aigaion.
- Automatisches Erkennen von Gerät (PC, Mobile Phone, etc.), damit die Seite entsprechend dargestellt wird.
- Implementierung von Schlagwortwolken (tag clouds), um die Kompetenzen

und Themen der einzelnen Autoren zu verdeutlichen. Bei der bereits bestehenden Möglichkeit Tag clouds zu integrieren, wird dies anhand von selber definierten Schlüsselwörtern realisiert. Das neue System soll die Schlagwortwolke automatisch anhand der veröffentlichten Arbeiten des Autors generieren.

2 Beschreibung des Ist-Zustands

2.1 Aigaion

Die Webseite <http://publications.hevs.ch> wurde mit Aigaion Version 2.1.2 realisiert.

Aigaion ist eine web-basierte Verwaltungssoftware für Bibliografien, welche frei verfügbar ist. Die Hauptfunktionen der Software sind:

- Auflistung der im System erfassten Publikationen und Autoren
- Unterteilung in Themen
- Exportieren ins BibTeX- und RIS-Format sowie formatierter Export nach gebräuchlichen Zitierstilen
- Benutzerverwaltung mit Rechteprofilen
- Erfassen, Editieren und Löschen von Publikationen

Die Webseite wurde mit PHP geschrieben und mit CSS Stylesheets ergänzt. Einige Funktionalitäten wurden ausserdem mit JavaScript und Ajax programmiert. Abbildung 1 zeigt die Startseite vor Implementierung der Erweiterungen.

Abbildung 1: Webseite <http://publications.hevs.ch>

The screenshot shows the website <http://publications.hevs.ch>. The header features the logo of HES-SO Valais Wallis, which includes the text 'Haute Ecole Spécialisée de Suisse occidentale', 'Fachhochschule Westschweiz', and 'University of Applied Sciences Western Switzerland'. A search bar is located in the top right corner.

The left sidebar contains a navigation menu with the following sections:

- BROWSE**
 - My Topics
 - My Bookmarks
 - All Topics
 - All Publications
 - All Authors
 - All Keywords
 - Unassigned
 - Recent
 - Search
- EXPORT**
 - Export all publications
- SORT BY**
 - Author
 - Title
 - Type/Journal
 - Year
 - Recently added
- NEW DATA**
 - New Publication
 - New Author
 - New Topic
 - Import
- SITE**
 - Help
 - About this site
 - Site Configuration
 - Site Maintenance
 - Manage All Accounts
- LOGGED IN:**
 - admin
 - My Profile
 - Topic Subscribe
 - Logout

At the bottom of the sidebar, there is a language selection menu: 'Deutsch, English, Nederlands, Norsk, <more...>'.

The main content area is titled 'All publications' and displays a list of publications. The first row shows a range of page numbers: '1-50 | 51-100 | 101-150 | 151-200 | 201-250 | 251-300 | 301-350 | 351-400 | 401-450 | 451-500 | 601-650 | 651-700 | 701-750 | 751-800 | 801-850 | 851-853 |'. Below this, the year '2010' is highlighted. The list of publications includes:

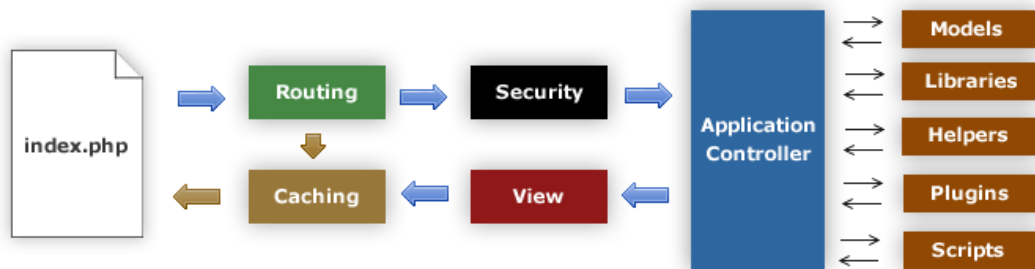
- 3D Case-Based Retrieval for Interstitial Lung Diseases**, Adrien Depeursinge, Alejandro Vargas, Alexandra Geissbuhler, Pierre-Alexandre Poletti and Henning Müller, in: MCBR-CDS 2009: Medical Content-based Retrieval for Clinical Decision Support, London, UK, pages 18-25, Springer, 2010
- A Comparative Analysis of Content in Traditional Survey versus Hotel Review Websites.**, Roland Schegg and Michael Fux, in: Information and Communication Technologies in Tourism 2010, pages 429-440, Springer-Verlag, New-York, 2010
- A web interface for 3D information retrieval with images from the lung**, Frédéric Gaillard, Adrien Depeursinge, Henning Müller, in: Medinfo 2010, Cape Town, South Africa, 2010
- An Investigation of Leapfrogging and Web 2.0 Implementation.**, Miriam Scaglione, Ahmad Fareed Ismail, Trabischet and Jamie Murphy, in: Information and Communication Technologies in Tourism 2010, SpringerWienNewYork, 2010
- An investigation of Motivation to Share Online Content by Young Travelers - Why and Where**, Hilary M. Roland Schegg, Springer-Verlag, Wien New-York (in press), 2010
- An investigation of Motivation to Share Online Content by Young Travelers - Why and Where.**, Hilary M. Roland Schegg, Springer-Verlag, Wien New-York, 2010
- Asymmetric margin support vector machines for lung tissue classification**, Jimison Tavindrasana, Adrien Depeursinge, Gilles Cohen, Antoine Geissbuhler and Henning Müller, in: WCCI 2010, Barcelona, Spain, 2010
- Comparative Performance Analysis of State-of-the-Art Classification Algorithms Applied to Lung Tissue Categorization**, Adrien Depeursinge, Jimison Tavindrasana, Asmaa Hidki, Gilles Cohen, Antoine Geissbuhler, Adrien Depeursinge, Pierre-Alexandre Poletti and Henning Müller (2010), in: Journal of Digital Imaging
- Content-based medical image retrieval**, Henning Müller and Thomas M. Desemo, 2010
- Die Bedeutung des Online-Vertriebs in der Schweizer Hotellerie.**, Roland Schegg and Michael Fux, hotelier, 2010
- Editorial of the special issue on image and video retrieval evaluation**, Allan Hanbury, Henning Müller and (2010), in: Computer Vision and Image Understanding, Special Issue on Image and Video Retrieval Evaluation
- Elites et mise en désir de lieux touristiques**, Rafael Matos-Wasem (2010), in: www.articulo.ch / Avant-garde de la société
- Evaluation campaigns in medical imaging**, Jayashree Kalpathy-Cramer and Henning Müller, 2010
- Exploring communication and marketing strategies for senior travelers**, Andrew Mungall, Roland Schegg and Michael Fux, 2010
- Fureur et vacarme dans les Alpes**, Rafael Matos-Wasem (2010), in: L'Alpe
- Fusing visual and clinical information for lung tissue classification in HRCT data**, Adrien Depeursinge, Adrien Depeursinge, Jimison Tavindrasana, Gilles Cohen, Alexandra Platon, Pierre-Alexandre Poletti and Henning Müller (2010), in: Artificial Intelligence in Medicine
- Health Data Depersonalisation for Prospective research in the life sciences**, Bernice Elger, Jimison Tavindrasana, Henning Müller, Nicolas Roduit, Paul Summers and Jessica Wright (2010), in: Computer Methods in Biomedicine
- ImageCLEF - Experimental Evaluation of Visual Information Retrieval**, Henning Müller, Paul Clough, Thoralf Zeller and Barbara Caputo (2010), in: Springer

Quelle: Eigene Darstellung

2.2 CodeIgniter

CodeIgniter basiert auf dem Open Source PHP Framework CodeIgniter (CI) [3]. Aufgrund des Model-View-Controller (MVC) Patterns bietet es einen sehr strukturierten Aufbau.

Abbildung 2: Funktionsweise CodeIgniter



Quelle: http://codeigniter.com/user_guide/images/appflowchart.gif

Abbildung 2 zeigt die Funktionsweise von CI. Die Datei `index.php` dient als Front-Controller und initialisiert die Basis Ressourcen, welche benötigt werden, damit die Seite funktionieren kann. Der Router analysiert dann den HTTP-Request des Users und entscheidet, was damit geschehen soll. Existiert ein Cache-File, wird der Request direkt an den Browser weitergeleitet. Ansonsten wird die Anfrage an das Sicherheitsmodul weitergeleitet, wo sie auf Gültigkeit überprüft wird. Der Controller lädt dann die benötigten Ressourcen, um die Anfrage abzuarbeiten. Zu den Ressourcen gehören:

- **Models:** Hier befinden sich normalerweise Funktionen wie etwa das Abrufen von Informationen in der Datenbank. Bei Aigaion wurden diese Funktionen jedoch in die Libraries verschoben.
- **Libraries:** Enthalten primär Klassen und Funktionen zur Kommunikation mit der Datenbank. In Aigaion wird ein objektorientierter Ansatz gewählt, d.h. die abgerufenen Informationen (Publikationen, Autoren, etc.) werden als Objekte gespeichert und die Feldnamen in der Datenbank als deren Attribute.
- **Helpers:** Diese Hilfedateien enthalten verschiedene nützliche, vordefinierte Funktionen wie etwa das Erstellen eines Formulars.
- **Plugins:** Hier werden verschiedene Erweiterungen gespeichert wie beispielsweise ein RTF-Konverter.
- **Scripts:** Zahlreiche JavaScripts werden hier platziert und können vom Controller geladen werden.

Schlussendlich wird die View (Präsentationsschicht) vom Controller geladen und an den Webbrowser gesendet. Die View wird ausserdem zwischengespeichert, damit sie beim nächsten Aufruf direkt aus dem Cache ausgelesen werden kann.

URLs bei CI funktionieren immer nach dem gleichen Prinzip:

`base_url/index.php/class/function/ID`

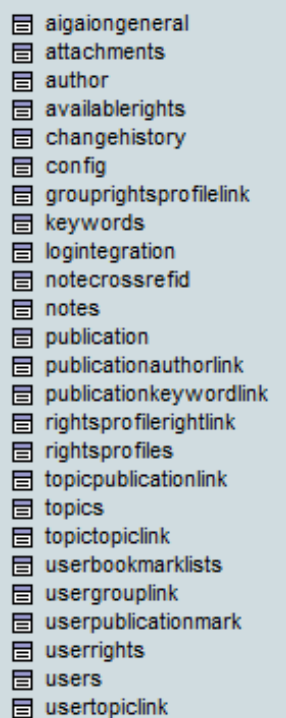
Mit dem ersten Segment ("class") wird der entsprechende Controller geladen. Im nächsten Segment wird eine Funktion in diesem Controller aufgerufen. Das dritte Segment gibt den Parameter an, der an die aufgerufene Funktion mitgegeben werden

kann. Auf Aigaion angewendet, wird z.B. mit der URL <http://publications.hevs.ch/index.php/publications/show/682> die Methode „show“ mit Paramater „682“ im Controller „publications“ aufgerufen. Im Browser wird dann die Publikation mit der ID 682 angezeigt.

2.3 Datenbank

Im Hintergrund steht eine MySQL-Datenbank zur Verfügung, in welcher die Daten gespeichert sind. Die Datenbank besteht aus 26 Tabellen (Abb. 3).

Abbildung 3: Datenbanktabellen



aigaiongeneral
attachments
author
availablerights
changehistory
config
grouprightsprofilelink
keywords
logintegration
notecrossrefid
notes
publication
publicationauthorlink
publicationkeywordlink
rightsprofilerightlink
rightsprofiles
topicpublicationlink
topics
topictopiclink
userbookmarklists
usergroupink
userpublicationmark
userrights
users
usertopiclink

Quelle: Eigene Darstellung

Dabei werden Hauptelemente wie Autor, Publikation, Thema und Mitglieder in eigene Tabellen gespeichert. Durch Verknüpfungstabellen kann danach ein Link zwischen den gewünschten Hauptelementen hergestellt werden (z.B. „topicpublicationlink“).

3 Analyse und Auswahl der Möglichkeiten

3.1 Export von Daten

Export nach Microsoft Excel


Für den Export der Bibliografien nach Excel gibt es grundsätzlich 3 Möglichkeiten.

Die erste Möglichkeit ist ein Export mittels eines PEAR-Packages, dem Spreadsheet Excel Writer. PEAR (PHP Extension and Application Repository) ist eine Bibliothek von Open Source Modulen und Erweiterungen für PHP [11]. Der Excel Writer erlaubt dabei Daten direkt in eine Excel-Tabelle zu schreiben. Die Implementierung dieses Packages wurde aus folgenden Gründen verworfen:

- Es handelt sich um ein relativ altes Package aus dem Jahr 2006 mit zahlreichen Fehlern. Der Entwickler selbst rät, dass das Package veraltet ist und ein komplettes Umschreiben des Codes nötig wäre.
- Es wird nur das BIFF5 Format unterstützt (Excel 5.0). Die fehlende Kompatibilität zum BIFF8 Format, welches neuere Excel Versionen benutzen, könnte Probleme verursachen.
- Es müssen zahlreiche Module wie PEAR oder OLE installiert werden.

Als zweite Möglichkeit wurde der Export in eine HTML-Tabelle in Betracht gezogen. Da Excel ohne Probleme auch HTML-Tabellen lesen kann (Abb. 4), muss lediglich eine solche Tabelle mit PHP generiert werden.

Abbildung 4: HTML-Tabelle in Excel dargestellt



```
<table>
<tr>
<td>2010</td>
<td>Hans Muster</td>
<td>Titel</td>
</tr>
<tr>
<td>2010</td>
<td>John Doe</td>
<td>Mustertitel</td>
</tr>
</table>
```

	A	B	C
1	2010	Hans Muster	Titel
2	2010	John Doe	Mustertitel

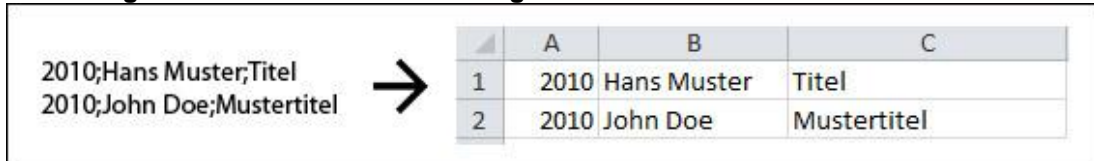
Quelle: Eigene Darstellung

Die Idee wurde jedoch verworfen, da aufgrund der zahlreichen HTML-Tags eine relativ grosse Datei erstellt wird.

Bei der dritten Möglichkeit soll der Export mittels Erstellen einer CSV-Datei durchgeführt werden. CSV (Comma-separated values) ist ein Dateiformat, in welchem Daten getrennt durch Komma oder Semikolon gespeichert werden können. Excel kann diese Dateien direkt öffnen oder importieren. Dabei bedeutet das Trennzeichen, dass

in Excel eine neue Spalte gebraucht werden soll. Eine neue Zeile in der CSV-Datei bedeutet für Excel auch eine neue Zeile (Abb. 5).

Abbildung 5: HTML-Tabelle in Excel dargestellt



	A	B	C
1	2010	Hans Muster	Titel
2	2010	John Doe	Mustertitel

Quelle: Eigene Darstellung

Aufgrund der einfachen Struktur des CSV-Dateiformats wurde diese Möglichkeit für den Export nach Excel gewählt.

Export für RERO

Nach Kontaktaufnahme mit dem Westschweizer Bibliotheksverbund RERO war klar, dass nur ein Export nach MARCXML in Frage kommt, da dies das bevorzugte Format für den Import nach RERO ist. MARCXML ist ein XML Schema zum einfachen Austausch von bibliografischen Informationen.

3.2 Generierung von Statistiken

Zur Generierung von Statistiken wurde eine Library gesucht, die mittels PHP Grafiken erstellen kann. Dabei wurde anhand folgender Kriterien gesucht:

- Grafiken müssen anpassbar sein, d.h. X- und Y-Achse des Graphen sollen veränderbar sein, damit verschiedene Zeitspannen ausgewählt werden können.
- Eine reibungslose Anbindung an die Datenbank muss gegeben sein, damit die Informationen aus der Datenbank dargestellt werden können.
- Verschiedene Diagrammtypen wie Balkendiagramm oder Liniendiagramm sollen erstellt werden können.
- Die Lösung muss Open Source sein.

Es wurden drei Bibliotheken ausgewählt und näher analysiert.

Flot

Flot ist eine Open Source Library, welche Grafiken mit Hilfe von JavaScript generiert [10]. Das Design der Graphen ist dabei sehr ansprechend. Dank JavaScript sind die generierten Grafiken sehr dynamisch, d.h. es können verschiedene Punkte des Graphen per Mausklick ausgewählt werden oder es kann herangezoomt werden. Nachteil von Flot ist die mangelnde PHP-Schnittstelle. Die ganze Library wurde ausschliesslich mit JavaScript programmiert, was eine Implementierung in die vorhandene Webseite erschwert. Des Weiteren erzeugt es keine traditionelle Grafiken im jpg- oder png-Format, sondern zeigt den Graphen nur dynamisch im HTML-Code eingebettet an.

GraPHPite

Die PHP-Bibliothek GraPHPite [5] kann eine Vielzahl von Diagrammtypen erstellen. Dabei werden die Graphen mittels GD Library erstellt, welche die Schnittstelle zwischen PHP und den Grafiken darstellt. GraPHPite bietet eine Vielzahl von Funktionen zur Anpassung der Diagramme. Leider besteht das Paket aus sehr vielen Dateien und ist damit sehr gross.

Jpgraph

Jpgraph ähnelt GraPHPite sehr. Es benutzt ebenfalls die GD Library zur Erstellung von Graphen, erstellt jedoch die optisch ansprechenderen Graphen, welche zudem durch eine Vielzahl von Funktionen anpassbar sind. Jpgraph hat die Möglichkeit, wahlweise Grafiken im jpg-, gif-, oder png-Format zu erstellen. Diese Funktionalitäten und die ausführliche HTML-Dokumentation [8] waren schlussendlich der Grund, die Statistiken mit Hilfe von Jpgraph zu implementieren.

3.3 Erhöhung der Sichtbarkeit

URL Rewriting

Durch ein Umschreiben der URLs soll in erster Linie eine Suchmaschinenoptimierung erreicht werden, d.h. die Webseite und deren Inhalt sollen mit Suchmaschinen wie zum Beispiel Google schneller gefunden werden. Zweitens sollen die URLs aussagekräftiger werden. Statt der ID von Publikationen, Themen und Autoren soll deren Namen in der URL angezeigt werden. Ausserdem sollen wenn möglich überflüssige Segmente in der URL gestrichen werden. Dazu wurden zwei Möglichkeiten in Betracht gezogen.

Erstens kann ein Rewriting mittels einer htaccess-Datei erreicht werden. Mit dieser Konfigurationsdatei können gewisse Regeln für die Webseite definiert werden und auch die URLs intern umgeschrieben werden. Externe HTTP-Anfragen würden dabei immer zuerst über diese Datei umgeleitet, in der nun die URL umgeschrieben wird und das Ergebnis zurückgegeben wird. Mit dieser Möglichkeit wurde jedoch nur das Entfernen der index.php in der URL erreicht. Die jeweilige ID durch den Namen zu ersetzen wäre sehr kompliziert, da aufgrund der Struktur von Aigaion die ID explizit in der URL stehen muss, damit das jeweilige Objekt aus der Datenbank ausgelesen werden kann.

Glücklicherweise stellt das CI Framework eine Router-PHP-Datei zur Verfügung, die ähnlich wie die .htaccess-Datei funktioniert, jedoch wesentlich komfortabler in der Konfiguration ist. Diese Methode wurde gewählt, um den Namen des jeweiligen Objekts in der URL anzeigen zu lassen und gewisse Segmente zu löschen.








Mailing Lists

Die ursprüngliche Idee der Mailing Lists war, dass registrierte Mitglieder automatisch eine E-Mail erhalten, sobald dem System eine neue Publikation hinzugefügt wird. Dies war jedoch nicht optimal, da der Datenbank ein neues Feld hätte hinzugefügt werden müssen, das angibt, ob das Mitglied per Mail benachrichtigt werden soll oder nicht. Da Aigaion aber eine Option zur automatischen Sicherung und Wiederherstellung der Datenbank besitzt, wäre es fraglich gewesen, ob bei Änderung der Datenbank diese Funktion noch korrekt funktioniert hätte.

Es wurde eine optimale Lösung angestrebt: Die Webseite besitzt bereits eine Funktion, bei der Themen abonniert werden können, welche dann unter „Meine Themen“ angezeigt werden können. Nun sollte eine Funktion eingebaut werden, die eine E-Mail an alle Abonnenten eines Themas sendet, sobald eine Publikation dem Thema zugeordnet wird. Das hat den Nachteil, dass nun niemand eine E-Mail erhält, wenn eine Publikation hinzugefügt, aber keinem Thema zugeordnet wurde. Dieser Umstand ist jedoch vernachlässigbar, da die Mitglieder hauptsächlich an neuen Publikationen ihrer abonnierten Themen interessiert sind.

4 Implementierung der Erweiterungen

4.1 Verwendete Software

Adobe Dreamweaver CS5	
	Mit dem HTML-Editor Dreamweaver wurden der HTML- und PHP-Code erzeugt. Durch seinen integrierten FTP-Client konnten die erstellten und veränderten Dateien direkt auf den Server geladen werden.
Linux Enterprise Server 10 SP2 als VMware Image	
	Als Webserver diente ein Linux Enterprise Server. Dazu wurde ein VMware-Abbild des originalen Servers erstellt, welches die Grundlage für die Entwicklung der neuen Funktionen darstellte.
Vsftpd 2.3.0	
	Der FTP-Server wurde auf dem Linux Server installiert, damit Dateien bequem per Dreamweaver hochgeladen werden konnten.
Windows 7 x64 Professional	
	Das neueste Betriebssystem von Microsoft diente als Client und Entwicklungsstation für die Erweiterungen.
XML Marker 1.1	
	Mit diesem einfachen Editor wurde die XML-Datei für den CSV-Export erstellt, sowie die Struktur der erzeugten MARCXML-Datei kontrolliert.
PoEdit	
	Hiermit konnten die vorhandenen Sprachdateien (.po), welche die Webseite für die Übersetzung braucht, geöffnet und editiert werden.
Verschiedene Browser	
	Die Webseite wurde laufend mit Firefox 3.6.8 getestet. Zum Schluss wurden die neuen Funktionen noch mit folgenden Browsern getestet: IE 8.0, Chrome 5.0, Opera 10.60, Safari 5.0.1

4.2 Modifikationen am Linux Server

Damit gewisse Erweiterungen überhaupt funktionierten, mussten eine Reihe von Änderungen am SUSE Linux Server angebracht werden.

Die Library jgraph benötigte zum Darstellen von Graphen die GD Bibliothek für PHP 5. Diese Bibliothek erlaubt das Generieren von Grafiken mit Hilfe von PHP-Funktionen. Das Paket php5-gd wurde mit Hilfe des Yast Software Managers installiert. Anschliessend wurde der Webserver mit „`/etc/init.d/apache2 restart`“ in der Konsole neugestartet.

Jgraph benötigt ausserdem einige TrueType-Schriften, um Text in den Graphen darstellen zu können. Die benötigten Schriften wurden aus Windows 7 kopiert und in das Verzeichnis „`/usr/X11R6/lib/fonts/truetype`“ auf dem Server abgelegt.

Damit ein Verändern der URLs mittels `.htaccess`-Datei möglich wurde, musste zuerst das Apache-Modul `mod_rewrite` aktiviert werden. Dazu wurde „`rewrite`“ in der Datei „`/etc/sysconfig/apache2`“ bei den zu ladenenden Modulen „`APACHE_MODULES=`“ hinzugefügt. Danach wurde in der Datei „`/etc/apache2/vhosts.d/aigaion2.conf`“ beim XML-Tag `<Directory "/srv/www/vhost/aigaion2"> AllowOverride` auf „`All`“ gesetzt, was bedeutet, dass das Setzen aller Direktiven in der `.htaccess` erlaubt ist.

4.3 Export von Daten

CSV Export

Beim Export gibt es zwei Arten zu unterscheiden, die schon implementiert worden sind. Erstens sind das die zwei Bibliografie-Formate BibTeX und RIS, deren Export durch eigene Funktionen realisiert worden ist. Zweitens ist ein formatierter Export nach HTML, TXT oder RTF möglich, bei dem verschiedene Zitierstile ausgewählt werden können (Abb. 6). Dieser Export wurde mit Hilfe dem PHP Plugin OSBiB 3.0 realisiert.

Abbildung 6: Export vor Erweiterung

Export all publications

Please select the format in which you want to export the publications:

BibTeX

RIS

Format: HTML Style: APA Export

Quelle: Eigene Darstellung

OSBiB benutzt dabei für jeden Zitierstil eine eigene XML-Datei, in der die Formatierung festgelegt wird. Ausserdem kann für jeden Publikationstyp (Artikel, Buch, etc.) beschrieben werden, welche Daten exportiert werden sollen und wodurch diese Daten getrennt werden sollen. Dies ist optimal für den CSV Export, da das Semikolon bequem im XML definiert werden kann.

Für den CSV Export wurde also eine XML-Datei eines existierenden Stils kopiert und verändert. Der neue Stil wurde „DATETITLEAUTHORATYPE“ genannt, denn die Struktur und die Informationen sollten folgendermassen exportiert werden:

„Monat;Jahr;Titel der Publikation, veröffentlicht in; Typ der Publikation“

Das entspricht in Excel einer Breite von vier Spalten.

Abbildung 7: Auszug „DATETITLEAUTHORATYPE.xml“

```
<resource name = "journal_article">
  <month>
    <pre></pre>
    <post></post>
  </month>
  <publicationYear>
    <pre></pre>
    <post></post>
  </publicationYear>
  <title>
    <pre></pre>
    <post></post>
  </title>
  <journal>
    <pre></pre>
    <post></post>
  </journal>
  <author>
    <pre></pre>
    <post></post>
  </author>
  <pubtype>
    <pre></pre>
    <post></post>
  </pubtype>
</resource>
```

Quelle: Eigene Darstellung

In dem in Abbildung 7 dargestellten Auszug der erstellten XML-Datei wird der Export eines Zeitungsartikels beschrieben. Die 5 XML-Tags auf zweiter Ebene entsprechen oben definierter CSV-Struktur. Dabei kann mit den Tags <pre> und <post> definiert werden, was vor resp. nach den Informationen ausgegeben werden soll. Für jeden Publikationstyp wurde eine solche Struktur definiert.

Nun mussten noch einige fehlende Elemente wie Monat und Publikationstyp in der Datei STYLEMAPBIBTEX.php eingetragen werden. Dort wird nämlich definiert, welche Informationen aus der Datenbank gelesen werden sollen. Bei den vorhandenen Stilen wurden diese zwei Elemente nicht gebraucht, darum weiss das OSBiB Plugin auch nicht, welches die entsprechenden Gegenstücke in der Datenbank sind. Der neu erstellte Stil wurde nun wie jeder andere Stil mit einer PHP-Funktion analysiert und die benötigten Information aus der Datenbank abgerufen.

Im Dropdown-Menü wird der Stil nun automatisch geladen und korrekt weiterverarbeitet. Was nun noch fehlte, war ein Export in eine CSV-Datei. Es wurde nun zuerst manuell ein Eintrag „CSV“ im „Format-Dropdown“-Menü erstellt. Diesem

Eintrag wurde nun die korrekte Dateieindung sowie der MIME-Typ „text/csv“ zugewiesen, damit der Browser die Datei korrekt interpretieren kann.

Nun sollten noch zwei Dropdown-Menüs implementiert werden, mit denen das Jahr der zu exportierenden Publikationen ausgewählt werden kann. Zudem kann eine Sortierung nach BibTeX ID, Jahr, Titel oder Publikationstyp vorgenommen werden.

Die einzelnen Jahre werden mit Hilfe einer MySQL-Abfrage dynamisch ins Dropdown-Menü geladen, d.h. es werden nur Jahre geladen, zu welchen es auch Publikationen gibt. Für die Sortierung wurde manuell ein Array erstellt.

Für das korrekte Auslesen mit den zwei neuen Filtern mussten nun neue Datenbank-Abfragen erstellt werden, die das Jahr und die Sortierung berücksichtigten. Schlussendlich wurden mit einer Schleife alle Publikationen durchlaufen und die entsprechende Methode aufgerufen, welche diese richtig formatiert. Nach Fertigstellung wird die Datei, je nach Benutzereinstellungen, entweder im Browser darzustellen versucht oder es wird eine Aufforderung angezeigt, die Datei herunterzuladen.

Der neue Stil funktioniert nun wahlweise für einen Export mit allen Publikationen, einem Autor, einem Thema oder einer einzelnen Publikation. Dies funktioniert wie gehabt durch Auswählen von z.B. Autor und Klicken auf „Export author“ anstelle von „Export all publications“ im Menü (Abb. 7).

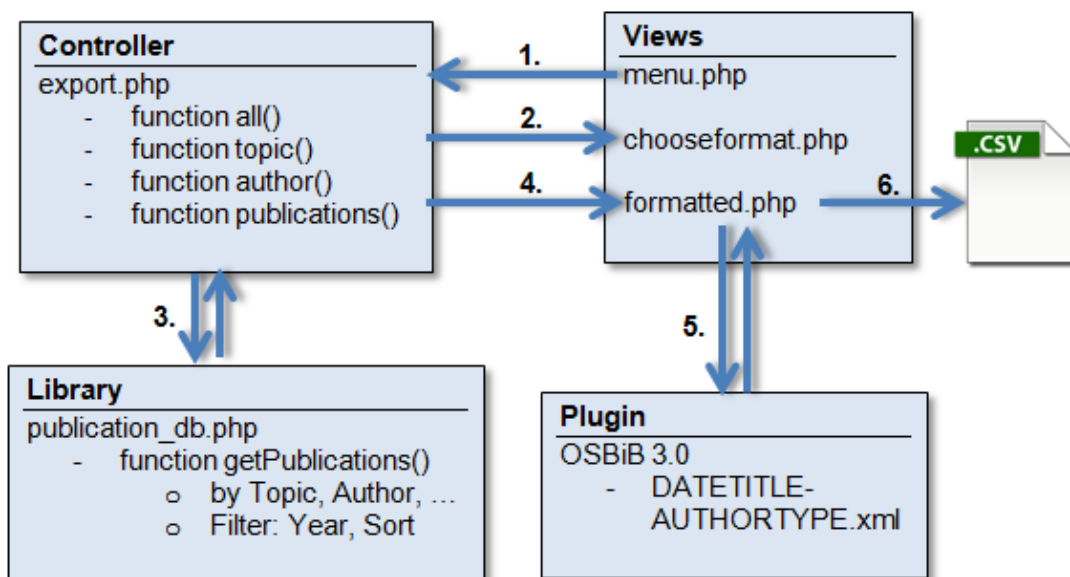
**Abbildung 8:
Export eines Autors**



Quelle: Eigene Darstellung

Die zwei Filter „Sortieren“ und „Jahr“ wurden auch für den vorhandenen Export nach BibTeX und RIS zur Verfügung gestellt, da diese ohne Probleme darauf angewendet werden konnten.

Abbildung 9: Vereinfachtes Schema CSV-Export



Quelle: Eigene Darstellung

Abbildung 9 zeigt den Ablauf des CSV Exports. Sobald der User auf einen Link im

Menü klickt, wird er zu der entsprechenden Funktion im Controller weitergeleitet. Klickt er zum Beispiel auf „Export author“, wird die Funktion „author()“ aufgerufen. Diese Funktion überprüft, ob das Formular zum Export schon ausgefüllt wurde. Wenn nicht, wird der User zum Auswahl-Bildschirm weitergeleitet (Abb. 10). Ansonsten (nach Klicken auf den Export-Button) liest der Controller die benötigten Publikationen aus der Datenbank. Danach ruft er die Datei formatted.php auf, in welcher der Output mit Hilfe von OSBiB erstellt wird und in eine CSV-Datei exportiert wird.

Abbildung 10: Export nach CSV-Erweiterung

Quelle: Eigene Darstellung

Leider benutzt Excel das falsche Encoding beim direkten Öffnen der CSV-Datei, weswegen Sonderzeichen nicht richtig dargestellt werden. Darum muss die Datei in Excel importiert werden. Dies erfolgt in Excel 2007/2010 durch das Excel-Ribbon „Daten“ und den Menüpunkt „Aus Text“.

MARCXML Export

Die Struktur der zu generierenden XML-Datei wurde durch RERO festgelegt (Abb. 11). In einem 15-seitigen PDF [14] werden die freiwilligen und obligatorischen XML-Tags beschrieben. Je nach Publikationstyp unterscheiden sich die Tags.

Abbildung 11: Auszug MARCXML Struktur RERO

```
<collection xmlns="http://www.loc.gov/MARC21/slim">
  <record>
    <datafield tag="980" ind1=" " ind2=" ">
      <subfield code="f">ART_JOURNAL</subfield>
    </datafield>
    <datafield tag="269" ind1=" " ind2=" ">
      <subfield code="c">2004</subfield>
    </datafield>
  </record>
</collection>
```

Quelle: <http://doc.rero.ch>

Der Tag <record> bezieht sich auf eine Publikation. Es können sich beliebig viele Publikationen in einer <collection> befinden. Das Attribut „tag“ von <datafield> beinhaltet eine vordefinierte Nummer. „980“ steht zum Beispiel für den Typ der Publikation. In den Tags <subfield code=„Attribut“> werden die eigentlichen Informationen festgehalten.

Ein MARCXML Export wurde vom Idiap Research Institute auf ihrer Publikationsseite bereits eingebunden und wurde freundlicherweise zur Verfügung

gestellt [7]. Dieser Export war jedoch nicht für RERO gedacht. Deshalb mussten die zu erstellenden XML-Tags fast komplett umgeschrieben werden.

Für die Erstellung der XML-Datei wurde ein objektorientierter Ansatz gewählt. Ein XML-Element wurde als Klasse definiert, die eine Methode zum Hinzufügen eines neuen XML-Tags und eine Methode zum Hinzufügen dieses Tags zu einem String besitzt. In der Hauptfunktion, welche eine Publikation zu einem XML umwandelt, wird nun aus dieser Klasse ein Objekt erstellt. Durch die add()-Funktion, welche als Parameter die Nummer, den Code und den Inhalt aufnimmt, können neue XML-Tags erstellt und diese dann in einem String gespeichert werden.

RERO setzt zwingend Tags für die Sprache der Publikationen voraus. Bei Aigaion gibt es leider keine Möglichkeit die Sprache festzulegen und die im System erfassten Publikationen besitzen somit alle keine Sprache. Deshalb wurde die Google Language Detection API zu Hilfe genommen, um die Sprache automatisch dem Titel der Publikation zu entnehmen [2, 4].

Abbildung 12: Funktion zur Spracheerkennung

```
//detect the language from a string using google API (used in marc21.php)
function detect_language($text)
{
    $version = '1.0';
    $url = 'http://www.google.com/uds/GlangDetect?v='.$version.'&sq='.urlencode($text);
    $response = json_decode(file_get_contents($url), true);

    if ($response['responseStatus'] == 200)
    {
        switch($response['responseData']['language'])
        {
            case 'en':
                return 'eng';
                break;
            case 'fr':
                return 'fre';
                break;
            case 'de':
                return 'ger';
                break;
            default:
                return '';
        }
    }
    else
        return '';
}
```

Quelle: Eigene Darstellung

Die Funktion in Abbildung 12 findet automatisch die Sprache eines Strings, in diesem Fall den Titel einer Publikation. Die URL mit dem zu erkennenden String liefert folgenden JSON (JavaScript Object Notation) zurück:

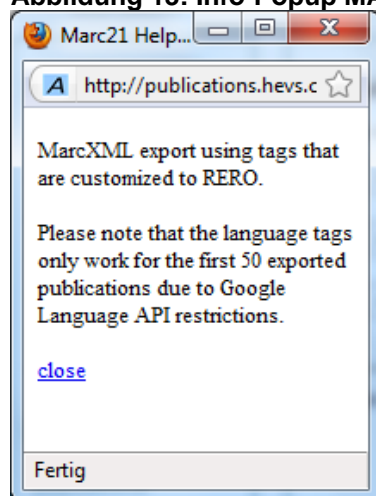
```
{"responseData":
{"language":"en","isReliable":false,"confidence":0.007940573},
"responseDetails": null, "responseStatus": 200}
```

Die zwei entscheidenden Werte sind „language“ und „responseStatus“. Letzterer liefert „200“ zurück, falls alles in Ordnung ist.

Mit der PHP-Funktion `json_decode()` werden nun diese Werte in ein mehrdimensionales assoziatives Array gespeichert. Bei erfolgreichem Status wird die Sprache im RERO-Format zurückgegeben, ansonsten ein leerer String.

Google hat seinen Dienst eingeschränkt. Sobald eine gewisse Anzahl von Anfragen schnell hintereinander gesendet wird, wird die IP des Aufrufers für kurze Zeit geblockt und eine Fehlermeldung angezeigt. Deswegen wurde die Spracherkennung in der PHP-Funktion auf die ersten 50 Publikationen begrenzt. Für die folgenden Publikationen wird jeweils ein leerer XML-Tag eingefügt.

Abbildung 13: Info-Popup MARC21



Quelle: Eigene Darstellung

Hierzu wurde ein Link neben dem Export-Button platziert, durch welchen ein Popup mit Erklärungen zum MARCXML-Export geöffnet wird (Abb. 13).

Abbildung 14: Export nach CSV- und MARCXML-Erweiterung

Export all publications

Please select the format in which you want to export the publications:

Year: Sort:

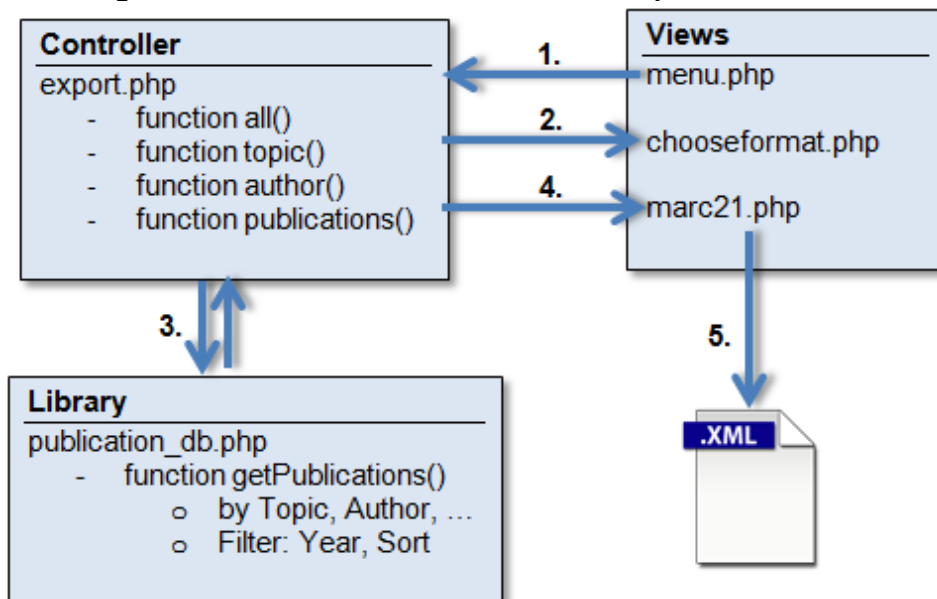
Year: Sort:

Year: Sort: [Info](#)

Format: Style: Year: Sort:

Quelle: Eigene Darstellung

Abbildung 14 zeigt das finale Auswahl-Menü des Exports. Der MARCXML-Export wurde ebenfalls mit den beiden Filtern „Jahr“ und „Sortieren“ ergänzt.

Abbildung 15: Vereinfachtes Schema MARCXML-Export

Quelle: Eigene Darstellung

Der MARCXML-Export funktioniert nach dem gleichen Prinzip wie der CSV-Export mit dem Unterschied, dass kein Plugin benötigt wird. Die Datei marc21.php ist für die Erstellung und Ausgabe der XML-Datei verantwortlich (Abb. 15).

4.4 Generierung von Statistiken mit jppgraph

Die Bibliothek Jppgraph in der Version 3.0.7 wurde als Plugin definiert [9]. Es enthält eine Vielzahl von Klassen zur Erstellung von Graphen, darunter auch viele Beispieldateien, in denen schon zahlreiche Graphen erstellt wurden. Es wurde eine neue Datei charts.php mit den Hauptfunktionen zur Generierung folgender 5 Graphentypen erstellt: Balkendiagramm, Liniendiagramm, gefülltes Liniendiagramm, Ballondiagramm (gross und klein) und Kreisdiagramm.

Abbildung 16: Funktion zur Erstellung eines Balkendiagramms

```

function barchart($datay, $datax, $title='Bar Chart')
{
    $graph = new Graph(700,500);
    $graph->img->SetMargin(60,20,35,75);
    $graph->SetScale("textlin");
    $graph->SetMarginColor("#BBD5EE");
    $graph->SetShadow();

    // Set up the title for the graph
    $graph->title->Set($title);
    $graph->title->SetMargin(8);
    $graph->title->SetFont(FF_VERDANA,FS_BOLD,10);
    $graph->title->SetColor("black");

    // Setup font for axis
    $graph->xaxis->SetFont(FF_VERDANA,FS_NORMAL,8);
    $graph->yaxis->SetFont(FF_VERDANA,FS_NORMAL,8);

    // Use a slightly bigger yscale so the graph doesn't go out of bounds
    $graph->yaxis->scale->SetGrace(5,0);

    // Show 0 label on Y-axis (default is not to show)
    $graph->yaxis->ticks->SupressZeroLabel(false);

    // Setup X-axis labels
    $graph->xaxis->SetTickLabels($datax);
    $graph->xaxis->SetLabelAngle(50);

    // Create the bar pot
    $bplot = new BarPlot($datay);
    $bplot->SetWidth(0.6);

    // Setup the values that are displayed on top of each bar
    $bplot->value->Show();

    $bplot->value->SetFormat('%01.00f');

    // Setup color for gradient fill style
    $bplot->SetFillGradient("#007AA4", "#BBD5EE", GRAD_HOR);

    $bplot->SetFillColor("#007AA4");

    // Set color for the frame of each bar
    $bplot->SetColor("white");
    $graph->Add($bplot);

    return $graph;
}

```

Quelle: Eigene Darstellung

Die in Abbildung 16 gezeigte Funktion erstellt ein Balkendiagramm. Als Parameter werden die Daten der X-Achse (in diesem Fall die Zeitspanne), der Y-Achse (Anzahl Publikationen) und der Titel des Diagramms übergeben. In der Funktion wird zuerst ein neues Graphobjekt erstellt. Mit Hilfe dessen Funktionen werden Titel und Layout der Datenachsen bestimmt. Ausserdem werden die Daten als Array der X-Achse zugewiesen. Mit einem BarPlot-Objekt wird der eigentliche Graph erstellt, welcher die Daten der Y-Achse als Argument erhält. Mit einer Reihe von Funktionen kann nun noch das Aussehen des Graphen verändert werden. Schlussendlich wird der Graph zurückgegeben.

Damit der Graph auch aufgerufen werden kann, braucht es zuerst einen neuen Menüpunkt. Dieser wurde mit einem Oberpunkt „Statistiken“ und Unterpunkt „Publikationen“ realisiert (Abb. 17).

Abbildung 17:
Menüpunkt Statistiken



Quelle: Eigene Darstellung

Anschliessend wurde ein neuer Controller „statistics.php“ erstellt. Dieser Controller wird mit Menüpunkt „Publikationen“ direkt aufgerufen. Der Controller erstellt den Graphen mit Hilfe der Funktionen in „charts.php“ und speichert ihn im png-Format in einem beschreibbaren Ordner auf dem Server. Nun wird die View „publications.php“ vom Controller aufgerufen. Diese enthält ein Formular mit verschiedenen Filtern zur Anzeige eines Graphs (Abb. 18).

Abbildung 18: Auswahlformular zur Anzeige eines Graphs

 A screenshot of a web form titled 'Publication statistics'. It contains several input fields: 'Topic' with a dropdown menu showing 'Informatique de gestion'; 'Author' with a text input showing 'Abbet, Samuel'; 'Type' with a dropdown menu showing 'Article'; 'Start' with a dropdown menu showing '1994'; 'End' with a dropdown menu showing '2010'; and 'Chart' with a dropdown menu showing 'Bar Chart'. There is also a 'Display' button.

Quelle: Eigene Darstellung

Die Filter „Thema“, „Autor“, „Start-“, und „Endjahr“ werden dynamisch aus der Datenbank ausgelesen. Mit Start- und Endjahr können Anfangs- und Endzeitpunkt der anzuzeigenden Publikationen gewählt werden. Mit der PHP-Funktion `array_slice()` wird die X-Achse (Zeitspanne) zurechtgeschnitten. Dem User wird dabei untersagt, ein Startjahr auszuwählen, das grösser als das Endjahr ist, denn ansonsten könnte kein Graph erzeugt werden. Es wird stattdessen eine vordefinierte Fehlermeldung angezeigt.

Die Filter können nun benutzt werden, um Daten und Aussehen des Graphs zu verändern. Mit Hilfe eines JavaScript onlicks wird beim Wechseln eines Wertes im Dropdown-Menü das Formular automatisch abgesendet und der Graph aktualisiert. Für sehr alte Browser, die JavaScript noch nicht oder nur teils unterstützen, wurde noch ein „Display“-Button eingebaut, der die Graphen ebenfalls aktualisiert. Beim Aktualisieren holt die View den vom Controller gezeichneten Graph und zeigt ihn im Browser unter dem Auswahlformular an.

Damit dies funktioniert, müssen Controller und View stetig Daten austauschen. Sobald ein Wert im Dropdown-Menü ausgewählt wird, muss der Controller die

Informationen mit Hilfe von Datenbankfunktionen neu abrufen und den Graph zeichnen.

In den Datenbank-Klassen wurden dabei neue Funktionen erstellt, welche die Anzahl Publikationen pro Jahr errechnen, und zwar unter Berücksichtigung der obengenannten Filter.

Es werden dabei immer zwei Graphen angezeigt. Im ersten sind die Anzahl Publikationen pro Jahr zu sehen. Der zweite zeigt ebenfalls die Anzahl Publikationen pro Jahr, kumuliert jedoch jeweils die Publikationen vom Vorjahr.

Abbildung 19: Füllen der X- und Y-Achse mit Daten

```
$datax = $this->publication_db->getAllYears2();

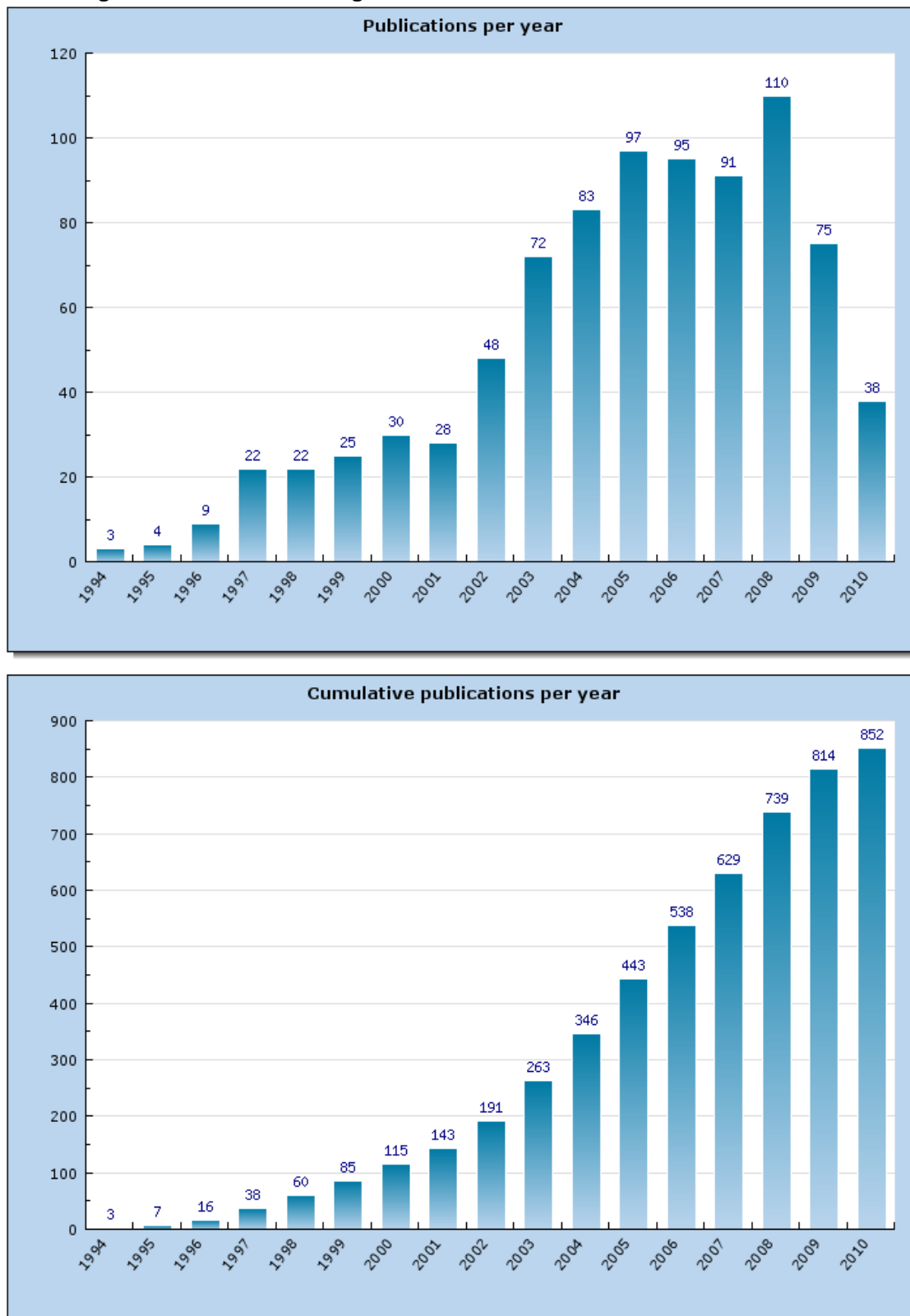
for ($i=0; $i < sizeof($datax); $i++)
{
    $datay[$i] = $this->publication_db->getPublicationCountForYear($datax[$i],
        $topic_id, $author_id, $pub_type);

    if($i==0)
        $sumdatay[$i] = $datay[$i];
    else
        $sumdatay[$i] = $sumdatay[$i-1] + $datay[$i];
}
```

Quelle: Eigene Darstellung

In Abbildung 19 wird gezeigt, wie die Daten der X- und Y-Achse abgerufen werden. Dabei handelt es sich bei beiden Achsen um Arrays. Zuerst wird die X-Achse mit Hilfe einer Datenbank-Funktion gefüllt, welche alle existierenden Jahre ausliest. Danach wird für jedes Jahr die Anzahl die Publikationen berechnet und in der Y-Achse gespeichert. Für die kumulierten Publikationen wurde ein eigenes Array definiert. Im ersten Jahr entspricht der Wert dem nicht kumulierten Wert, weil es noch kein Vorjahr gibt. Für alle folgenden Jahre werden die Werte addiert und jeweils im neuen Array gespeichert.

Abbildung 20 zeigt die beiden erstellten Graphen für alle Publikationen. In diesem Beispiel wurde für die Darstellung das Balkendiagramm gewählt.

Abbildung 20: Erstellte Balkendiagramme

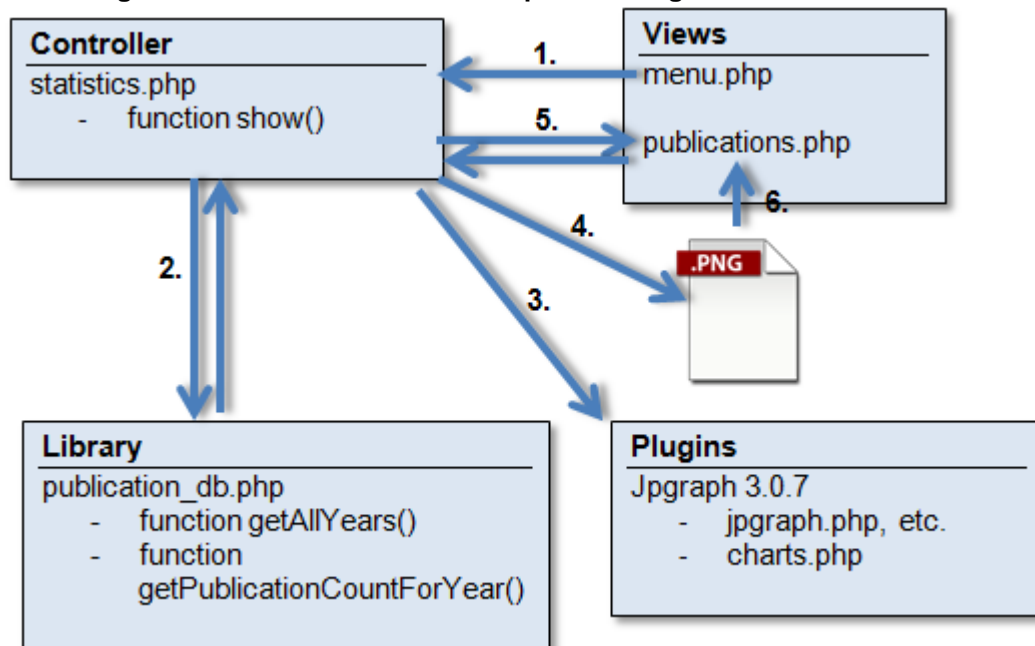
Quelle: Eigene Darstellung basierend auf www.jpgraph.net

Abbildung 21: Diagrammtypen

Quelle: Eigene Darstellung

In Abbildung 21 sind die vier weiteren Graphtypen dargestellt.

Der logische Ablauf zur Erstellung eines Graphen ist in Abbildung 22 dargestellt. Durch ein Klicken auf den entsprechenden Menüpunkt wird der Controller „statistics.php“ mit der Funktion show() aufgerufen. Dieser holt alle nötigen Datenbankfunktionen. Danach wird das Plugin geladen und der Graph mit Hilfe der entsprechenden Funktion in charts.php erstellt. Der Graph wird dann als png auf dem Server abgespeichert. Der Controller ruft nun die View publications.php auf. Die hier veränderten Werte in den Dropdown-Menüs werden an den Controller gesendet, welcher jedes Mal die gespeicherte Grafik überschreibt. Die Grafik wird dann in der View und schlussendlich im Browser angezeigt.

Abbildung 22: Vereinfachtes Schema Grapherstellung

Quelle: Eigene Darstellung

Nach einem Test mit allen Webbrowsern wurde festgestellt, dass der Internet Explorer trotz Auswahl von anderen Informationen in den Dropdown-Menüs immer die gleiche Grafik darstellt. Grund dafür war sein Cache. Der Browser ruft die Grafiken aus dem Zwischenspeicher ab. Die Grafiken müssten aber jedes Mal vom Server abgerufen werden. Trotz PHP-Header, welcher allen Browsern verbietet den Cache zu benutzen, hat der Internet Explorer immer seinen Zwischenspeicher gebraucht. Die Lösung war das Anfügen eines sich immer ändernden Parameters bei der Anzeige der Grafik mit HTML-Code. Als Parameter wurde der UNIX-Timestamp gewählt. Dieser gibt immer die aktuelle Zeit aus. Somit wird der Browser gezwungen, die Grafik bei jedem Abruf vom Server zu holen.

4.5 Erhöhung der Sichtbarkeit

URL Rewriting

Das URL Rewriting wurde in drei Schritten implementiert. Zuerst wurde die index.php in der URL entfernt. Dazu konnte in der CI Konfigurationsdatei der Wert `$config['index_page'] = 'index.php'` auf einen leeren String gesetzt werden. Dadurch wurde die „index.php“ nicht mehr in der URL angezeigt. Da aber die index.php essentiell für die Funktionsfähigkeit der Seite war, musste eine Weiterleitung implementiert werden. So musste zum Beispiel `http://publications.hevs.ch/publications` intern an `http://publications.hevs.ch/index.php/publications` weitergeleitet werden. Dies wurde mittels folgender Regel in der .htaccess-Datei realisiert:

```
RewriteCond $1 !^(index\.php|images|robots\.txt)
```

```
RewriteRule ^(.*)$ index.php?/$1 [L]
```

Die Regel besagt, dass alle HTTP-Anfragen ausser jenen für die „index.php“ selber, die Bilder und der robots.txt an die „index.php“ weitergeleitet werden sollen.

In einem zweiten Schritt wurde nun eine neue URL-Struktur für die Anzeige von Publikationen, Themen und Autoren festgelegt und die betroffenen URLs im Code ausgetauscht. Folgende Tabelle zeigt die alte sowie die neue Struktur:

Alte URL-Struktur
<code>http://publications.hevs.ch/index.php/publications/show/ID</code>
<code>http://publications.hevs.ch/index.php/authors/show/ID</code>
<code>http://publications.hevs.ch/index.php/topics/single/ID</code>
Neue URL-Struktur
<code>http://publications.hevs.ch/publication/ID/name_der_publication</code>
<code>http://publications.hevs.ch/author/ID/vorname_nachname</code>
<code>http://publications.hevs.ch/topic/ID/name_des_themas</code>

Die ID ist zwingender Bestandteil der URL und konnte nicht gestrichen werden, denn mit einer Funktion `getbyID()` wird das jeweilige Objekt aus der Datenbank gelesen. Eine Funktion `getbyName()` wäre nicht optimal gewesen, weil dann nicht zwei gleiche Namen in der Datenbank existieren könnten.

Das Objekt wird also wie gehabt aufgrund der ID aus der Datenbank abgerufen. Der Name konnte nun mit z.B. „`$publication->name`“ angezeigt werden. Da in URLs keine Sonderzeichen erlaubt sind, musste eine Funktion erstellt werden, die diese entfernt. So wurde aus „Jérôme Müller“ der URL konforme String „`jerome_mueller`“.

Da nun die neuen URLs noch nicht funktionierten, musste ein internes Weiterleiten auf die alte Struktur bewerkstelligt werden. CI bietet hierfür seine Routing-Klasse an (Abb. 23).

Abbildung 23: CI Routing

```
$route['author/(:num)/[a-zA-Z0-9\-\_]+'] = "authors/show/$1";
$route['author/(:num)/[a-zA-Z0-9\-\_]+/[a-z]+' ] = "authors/show/$1/$2"; //sort
$route['author/(:num)/[a-zA-Z0-9\-\_]+/[a-z]+/(:num)'] = "authors/show/$1/$2/$3"; //sort + multipage

$route['publication/(:num)/[a-zA-Z0-9\-\_]+'] = "publications/show/$1";
$route['publication/(:num)/[a-zA-Z0-9\-\_]+/[a-z]+' ] = "publications/show/$1/$2"; //categorize

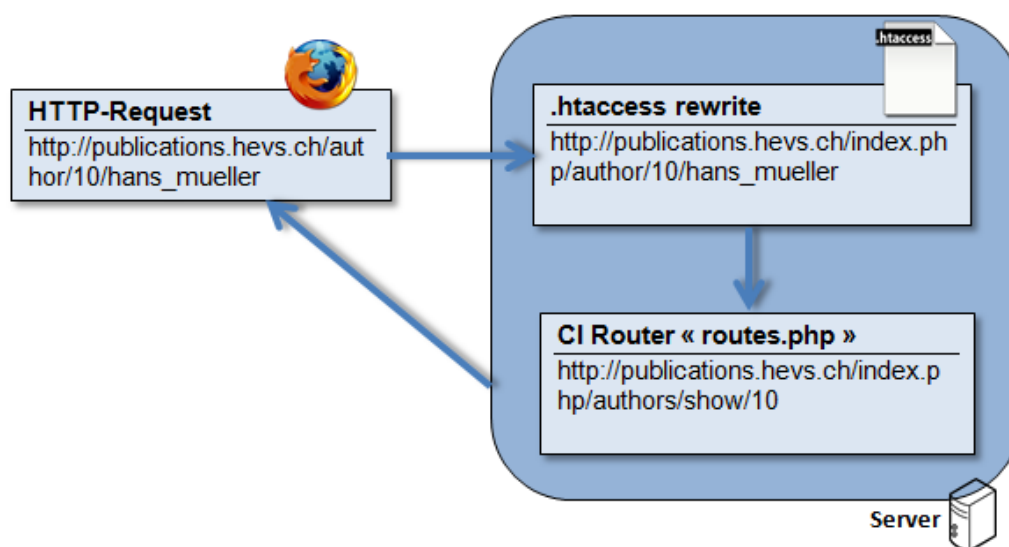
$route['topic/(:num)/[a-zA-Z0-9\-\_]+'] = "topics/single/$1";
$route['topic/(:num)/[a-zA-Z0-9\-\_]+/[a-z]+' ] = "topics/single/$1/$2"; //sort
$route['topic/(:num)/[a-zA-Z0-9\-\_]+/[a-z]+/(:num)'] = "topics/single/$1/$2/$3"; //sort + multipage
```

Quelle: Eigene Darstellung

Auf der linken Seite (Index des Arrays) wird die neue URL-Struktur festgelegt und auf

der rechte Seite (Wert) die alte Struktur, d.h. dort, wo intern weitergeleitet wird. Da es theoretisch unendlich viele IDs und Namen gibt, wurde mit Regular Expressions gearbeitet. Die erste Zeile bedeutet zum Beispiel, dass alle URLs, die im ersten Segment „author“ und im zweiten und dritten Segment irgendeine Zahl, Buchstaben oder Unterstrich besitzen, an „authors/show/\$1“ weitergeleitet werden. Der Wert \$1 bedeutet dabei, dass die erste Variable (:num) auf der linken Seite übernommen wird. Da es aufgrund des Sortierens und der Anzeige auf mehreren Seiten auch Links im Stil von „publication/ID/name/jahr/4“ gab, mussten diese auch korrekt umgeleitet werden. Der logische Ablauf der Weiterleitung wird in Abbildung 24 dargestellt.

Abbildung 24: Schema URL Rewriting



Quelle: Eigene Darstellung

Mailing Lists

Da die Implementierung von traditionellen Mailing Lists nicht optimal gewesen wäre, wurde ein dynamischer Ansatz gewählt. Die Mailing Lists werden anhand der Publikation und ihren verlinkten Themen automatisch erstellt.

Wie gehabt kann ein Mitglied, welches entsprechende Rechte besitzt, einer Publikation ein oder mehrere Themen zuweisen. Dies geschieht über den Link „categorize“ bei der Anzeige einer Publikation. Nun wird die Baumstruktur der Themen angezeigt und durch Klicken auf ein Thema wird dieses automatisch der Publikation zugewiesen. Dies geschieht mit Ajax. Deshalb muss die Seite nicht immer neu geladen werden (Abb. 25).

Abbildung 25: View show_publication.php

A web interface for 3D information retrieval with images from the lung [\[delete\]](#) [\[edit\]](#) [\[Bookmark\]](#) [\[E-mail\]](#) [\[BibTeX\]](#) [\[RIS\]](#) [\[Marc21\]](#) [\[Request\]](#)

Type of publication:	Inproceedings
Citation:	GDM2010
Booktitle:	Medinfo 2010
Year:	2010
Location:	Cape Town, South Africa
Keywords:	
Authors	Gaillard, Frédéric Depeursinge, Adrien Müller, Henning
Added by:	[]
Access rights:	r: e: (Edit all rights)
Total mark:	0
Your mark:	<input type="button" value="Read/Add mark"/> 1 5

Topics [\[finish categorization\]](#)

Click on a topic name to change its subscription status. Then click on "finish categorization".

- Economie d'entreprise
- ☒ Informatique de gestion
 - Accès ubiquitaire au Dossier Radiologique du patient
 - Adaptive Mobile and Market

Quelle: Eigene Darstellung

Durch Klicken auf „finish categorization“ wird nun eine E-Mail an alle Mitglieder geschickt, welche die entsprechenden Themen abonniert haben. Die E-Mail wird nur verschickt, falls die Mailing List nicht leer ist. Bei erfolgreichem Versand wird eine Meldung angezeigt. Falls Probleme beim Versand auftreten, wird eine Fehlermeldung angezeigt. (Abb. 26).

Abbildung 26: Meldungen nach E-Mail Versand

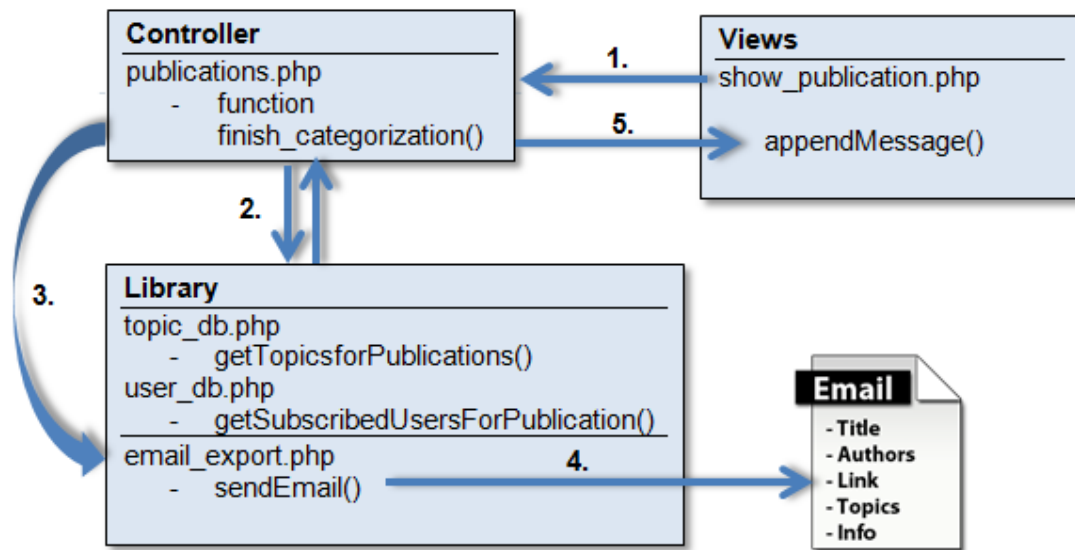
A notification email has been sent to the users who are subscribed to the affected topics.

The notification email could not be sent to the users who are subscribed to the affected topics. There might be something wrong with the users email addresses. Ask your admin to check the email addresses of every user (No email address is allowed too).

Quelle: Eigene Darstellung

Jedes Mitglied kann seine abonnierten Themen unter „Subscribe Topic“ verwalten und mit „My Topics“ anzeigen lassen.

Abbildung 27 Vereinfachtes Schema Mailing List



Quelle: Eigene Darstellung

In Abbildung 27 wird der Prozess des Emailversands dargestellt. Das Mitglied befindet sich zuerst in der View, auf welcher die jeweilige Publikation angezeigt wird. Durch Abschliessen der Zuordnung von Themen zu dieser Publikation wird im Controller die Funktion `finish_categorization()` aufgerufen. Diese Funktion holt nun alle Themen für diese Publikation (zur Anzeige in der E-Mail) sowie die Mitglieder, welche die entsprechenden Themen dieser Publikation abonniert haben. Daraus wird die Mailing Liste erstellt. An die Mitglieder dieser Liste wird nun mit der Funktion `sendEmail()` eine E-Mail verschickt. In der E-Mail sind folgende Informationen enthalten:

- Betreff: Eine neue Publikation wurde zu ihren abonnierten Themen hinzugefügt
- Titel der Publikation
- Autoren der Publikation
- Link zur Publikation
- Alle Themen, welche dieser Publikation zugeordnet sind
- Hinweise, wie neue Themen abonniert werden können oder wie sich das Mitglied von der Mailing Liste abmelden kann

Abschliessend ruft der Controller wieder die View auf und eine Meldung wird angehängt (Abb. 26).

4.6 Übersetzung

Die Webseite kann standardmässig neben Englisch noch auf Deutsch, Niederländisch und Norwegisch angezeigt werden. Damit der geschriebene Text der neu implementierten Funktionen nicht nur auf Englisch angezeigt wird, musste dieser noch übersetzt werden. Dabei wurde der Text nur auf Deutsch übersetzt.

Die Übersetzungen werden dabei als String in einer speziellen Datei gespeichert. Mit dem Tool PoEdit kann diese Datei geöffnet und editiert werden. So wurden alle neu hinzugefügten Wörter hier manuell übersetzt.

5 Schlussfolgerungen

5.1 Fazit

In dieser Arbeit wurde die bestehende Publikationsseite der HES-SO Sierre mit zusätzlichen Funktionalitäten versehen.

Dabei wurden ein CSV- und ein MARCXML-Export der Bibliografien realisiert. Es stellte sich heraus, dass ein CSV-Export der Daten nicht ohne Umweg funktioniert. Da Excel beim Öffnen einer CSV-Datei die falsche Zeichenkodierung benutzt, werden zum Beispiel Umlaute nicht richtig dargestellt. Das Problem kann umgehen werden, in dem die CSV-Datei mit der Importfunktion in Excel eingelesen wird. Der Inhalt wird auf diese Weise richtig dargestellt. Leider sind die Möglichkeiten begrenzt, einen direkten Export in eine Excel-Datei (.xls) zu bewerkstelligen.

Dann wurde zu Statistikzwecken eine automatische Generierung von Graphen implementiert, welche die Anzahl Publikationen pro Jahr anzeigen. Die Informationen werden dabei dynamisch aus der Datenbank geholt. Aufgrund eingebauter Filter wie beispielsweise das Auswählen der Zeitspanne sind diese Statistiken auch bestens für die Zukunft geeignet.

Um eine Suchmaschinenoptimierung zu erreichen, wurden abschliessend die URLs umgeschrieben. Ob dies die gewünschte Wirkung erzielt bleibt abzuwarten. Auch nach Migration der neuen Seite werden die Suchmaschinen einige Zeit brauchen, um die neue Struktur der Webseite indexieren zu können. Zu dieser letzten Funktionalität gehörte ausserdem noch das Erstellen von Mailing Lists. Durch das dynamische Erstellen einer solchen Mailing List wird eine E-Mail an alle Mitglieder versendet, die ein bestimmtes Thema abonniert haben. Dies geschieht sobald eine neue Publikation mit diesem Thema verlinkt wurde.

Somit wurden alle Hauptfunktionalitäten realisiert. Da das ausgiebige Testen dieser Erweiterungen auf deren korrekte Funktionsweise im Vordergrund stand und somit sehr viel Zeit in Anspruch nahm, konnte keines der optionalen Ziele realisiert werden.

5.2 Ausblick und Erweiterbarkeit

Die komplette Webseite mit Erweiterungen wird demnächst in das System eingebunden. Somit wird die alte Seite ersetzt. Es bleibt nur zu hoffen, dass die Migration problemlos verläuft. Zwar wurde die Seite extensiv getestet, doch eine vollständig fehlerfreie Software zu programmieren ist nahezu unmöglich.

Folgende potentielle Erweiterungsmöglichkeiten bieten sich für zukünftige Arbeiten an:

- Da die Struktur zur Erstellung einer XML-Datei festgelegt wurde, können nun nach einer kleinen Anpassung Daten in ein beliebiges XML-Schema exportiert

werden.

- Mit jpgraph ist eine leistungsstarke Bibliothek in das System integriert worden. Weitere Statistiken wie etwa die Anzeige der Anzahl Autoren sind dadurch leicht einzubinden.

Quellenverzeichnis

- [1] AIGAION (2010): *Aigaion - A Web based bibliography management software*, <http://www.aigaion.nl/> (Stand: 13.08.2010)
- [2] BLOMME, LODE (2008): *Server Side Language Detection with Google Language API*, <http://blog.lodeblomme.be/2008/03/21/server-side-language-detection-with-google-language-api/> (Stand: 04.08.2010)
- [3] CODEIGNITER (2010): *CodeIgniter User Guide Version 1.7.2*, http://codeigniter.com/user_guide/ (Stand: 12.08.2010)
- [4] GOOGLE (2010): *AJAX Language API for Translation and Detection*, <http://code.google.com/intl/de-DE/apis/ajaxlanguage/documentation/> (Stand: 04.08.2010)
- [5] GRAPHPITE (2010): *GraPHPite, Open Source PHP OO Graph Rendering Utility*, <http://graphpите.sourceforge.net/> (Stand: 11.06.2010)
- [6] HES-SO (2010): *Publications HES-SO*, <http://publications.hevs.ch> (Stand: 13.08.2010)
- [7] IDIAP Publications (2010): *Export Marc21*, <http://publications.idiap.ch/index.php/export> (Stand: 05.08.2010)
- [8] JPGRAPH (2010): *Documentation*, <http://jpgraph.net/download/manuals/> (Stand: 23.07.2010)
- [9] JPGRAPH (2010): *Download Version 3.0.7*, <http://jpgraph.net/download/> (Stand: 20.07.2010)
- [10] LAURSEN, OLE (2010): *flot, Attractive Javascript plotting for jQuery*, <http://code.google.com/p/flot/> (Stand: 06.08.2010)
- [11] LEVEFRE, FRANCK (2009): *PEAR Spreadsheet Excel Writer*, http://pear.php.net/package/Spreadsheet_Excel_Writer/ (Stand: 17.06.2010)
- [12] LIBRARY OF CONGRESS (2010): *MARCXML Standard*, <http://www.loc.gov/standards/marcxml/> (Stand: 23.07.2010)
- [13] PHP.NET (2010): *Allgemeine PHP Referenz*, <http://www.php.net/> (Stand: 06.08.2010)
- [14] RERO (2010): *Métadonnées de la bibliothèque numérique RERO DOC*, http://www.rero.ch/pdfview.php?section=fiche&filename=metadonnees_rerodoc.pdf (Stand: 23.07.2010)

Glossar

Ajax	Erlaubt eine asynchrone Übertragung zwischen Server und Browser. Somit können Funktionen aufgerufen werden, ohne die Webseite neu zu laden.
API	Schnittstelle zur Anwendungsprogrammierung
BibTeX und RIS	Formate zum Austausch von Bibliografien
BIFF5 / BIFF8	Von Excel eingesetzte Formate
Cache	Zwischenspeicher
CSS	Formatierungssprache für HTML-Dateien
CSV	Dateiformat, in welchem Daten getrennt durch Komma oder Semikolon gespeichert werden können.
Framework	Programmiergerüst, das als Basis für z.B. eine Webseite dienen kann.
FTP	Netzwerkprotokoll zur Übertragung von Dateien über IP-Netzwerke
GD Library	Programmbibliothek zur Erstellung von Grafiken
HES-SO	Haute Ecole Spécialisée de Suisse occidentale – Fachhochschule Westschweiz
HTTP	Protokoll zur Übertragung von Informationen aus dem Internet an einen Browser
ID	Identifikationsnummer
JavaScript	Skriptsprache zum Hinzufügen von erweiterten Anzeigeeffekten
JSON	JavaScript Datenformat zum Austausch von Informationen
Library	Programmbibliothek
Mailing List	E-Mail Versandliste
MARCXML	Ein XML-Schema, das vom Library of Congress entwickelt wurde, um den Austausch von Bibliografien zu erleichtern.
MySQL	Relationales Datenbankverwaltungssystem
OLE	Protokoll, dass die Zusammenarbeit unterschiedlicher Applikationen ermöglicht

Open Source	öffentlich zugänglicher Quelltext einer Software
Pattern	Entwurfsmuster, das für wiederkehrende Probleme verwendet werden kann
PHP	Skriptsprache, die zur Erstellung dynamischer Webseiten gebraucht wird.
Request	Anfrage
Rewriting	Umschreiben
String	Zeichenkette
URL	Uniform Ressource Locator – Internetadresse
Webbrowser	Software zum Anzeigen von Internetseiten
XML-Tag	Einzelnes Element in einer XML-Datei

Bestätigung

Ich bestätige hiermit, dass ich die vorliegende Bachelorarbeit alleine und nur mit den angegebenen Hilfsmitteln realisiert habe und dass ich ausschliesslich die erwähnten Quellen benutzt habe. Ohne Einverständnis des Studiengangleiters und des für die Bachelorarbeit verantwortlichen Dozenten sowie des Forschungspartners, mit dem ich zusammengearbeitet habe, werde ich diesen Bericht an niemanden verteilen, ausser an die Personen, die mir die wichtigsten Informationen für die Verfassung dieses Berichts geliefert haben.

Sierre, 16. August 2010

Samuel Werner

.....

Anhang

[illegible]

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen

Samuel Werner

Verantwortlicher Dozent : Jean Hennebert

Woche : 1
(17. Mai 2010)

Deklaration Wochenstunden

Aufgabe	Stunden
Einführung in das Projekt (Informationssitzung)	2
Sitzung mit J. Hennebert und H. Müller	1
Lesen von Bachelor-Dokumenten	4
Erstellung des Pflichtenhefts	4
TOTAL	11

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen

Samuel Werner

Verantwortlicher Dozent : Jean Hennebert

Woche : 2
(24. Mai 2010)

Deklaration Wochenstunden

Aufgabe	Stunden
Projektverwaltung	0.5
Sitzung mit J. Hennebert	1
Anfangsplanung	4
Suche von Möglichkeiten zur Generierung von Grafiken	8
TOTAL	13.5

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen

Samuel Werner

Verantwortlicher Dozent : Jean Hennebert

Woche : 3
(31. Mai 2010)

Deklaration Wochenstunden

Aufgabe	Stunden
Projektverwaltung	0.5
Sitzung mit H. Müller	1
Sitzung mit C. Pignat	1
Studieren/Verstehen der Website und Aigaion	16
TOTAL	18.5

Probleme / Lösungen

Beschreibung des Problems	Beschreibung der Lösung (wenn gefunden...)
Kein Login für die Webseite erhalten.	Md5-Hash vom admin-Passwort in der Datenbank in google eingegeben und den dazugehörigen originalen String gefunden.

Diverses

Fragen

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen

Samuel Werner

Verantwortlicher Dozent : Jean Hennebert

Woche : 4
(7. Juni 2010)

Deklaration Wochenstunden

Aufgabe	Stunden
Projektverwaltung	1
Studieren/Verstehen der Website Aigaion	8
Suche von Möglichkeiten zur Generierung von Grafiken	8
TOTAL	17

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen

Samuel Werner

Verantwortlicher Dozent : Jean Hennebert

Woche : 5
(14. Juni 2010)

Deklaration Wochenstunden

Aufgabe	Stunden
Projektverwaltung	1
Sitzung mit J. Hennebert	1
Suche Möglichkeiten zur Generierung von Grafiken	8
Suche Möglichkeiten zum Export von Daten	24
Abschlussdokumentation (Dokumentstruktur & Stichwörter)	2
TOTAL	36

Probleme / Lösungen

Beschreibung des Problems	Beschreibung der Lösung (wenn gefunden...)
GD-Graphics Library Fehlermeldung beim Test-Konvertieren eines Arrays in eine Grafik.	Installation des Pakets php5-gd über yast (Software Manager) und Neustart von apache2.

Diverses

Fragen

Welche Daten müssen mindestens grafisch dargestellt werden?

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen

Samuel Werner

Verantwortlicher Dozent : Jean Hennebert

Woche : 6
(5. Juli 2010)

Deklaration Wochenstunden

Aufgabe	Stunden
Projektverwaltung	1
Sitzung mit J. Hennebert	1
Implementierung Export von Daten	40
Studieren/Verstehen von Aigaion (CodeIgniter, XML)	2
Abschlussdokumentation (Stichwörter)	2
TOTAL	46

Probleme / Lösungen

Beschreibung des Problems	Beschreibung der Lösung (wenn gefunden...)
Excel Spaltenbreite wird nicht angepasst beim Öffnen einer csv-Datei.	Nicht lösbar. Spaltenbreite muss manuell angepasst werden.
Monat wird nur im folgenden Format dargestellt beim Export: „January“, „February“, etc.	
Chronologisch nach Monat sortieren eventuell schwer, da Monat als String gespeichert ist. („January“,...)	
Bei Publikationen vom Typ „Misc“ wird das Journal nicht angezeigt beim Exportieren.	Fehler behoben im <journal>-Tag in der XML-Datei.
Beim XML wird der Strichpunkt vor dem Autor nicht eingefügt, wenn der Titel mit einem Satzzeichen endet („“, „“, „?“).	Leerschlag vor dem Strichpunkt eingefügt.

Diverses

Export komplizierter als gedacht! ☹

Fragen

- Export-Filter nach Typ (Book, Article, ...)?
- Wie soll Autoren-Filter funktionieren? Wieso werden nicht alle Autoren aufgelistet, sondern immer nur 1-2.

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen

Samuel Werner

Verantwortlicher Dozent : Jean Hennebert

Woche : 7
(12. Juli 2010)

Deklaration Wochenstunden

Aufgabe	Stunden
Implementierung Export von Daten	8
Analyse Suchmaschinenoptimierung	8
Implementierung Generierung von Grafiken/Statistiken	24
Abschlussdokumentation (Stichwörter)	2
TOTAL	42

Probleme / Lösungen

Beschreibung des Problems	Beschreibung der Lösung (wenn gefunden...)
Nach Monat sortieren beim Export schwer, da Monat als String gespeichert ist. („January“,...). Sehr schwer umzusetzen auch im Parser-Funktion, da es auch Monate im Format „September/October“ oder „February15~17“ gibt.	

Diverses

Miguel Moreira von RERO hat meinem Email noch nicht geantwortet, betreffend Export ins RERO-Format (MARCXML).

Fragen

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen

Samuel Werner

Verantwortlicher Dozent : Jean Hennebert

Woche : 8
(19. Juli 2010)

Deklaration Wochenstunden

Aufgabe	Stunden
Projektverwaltung	1
Sitzung mit H. Müller	1
URL Rewriting	16
Implementierung Generierung von Grafiken/Statistiken	24
Abschlussdokumentation (Stichwörter)	1
TOTAL	43

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen

Samuel Werner

Verantwortlicher Dozent : Jean Hennebert

Woche : 9
(26. Juli 2010)

Deklaration Wochenstunden

Aufgabe	Stunden
Projektverwaltung	1
Sitzung mit J. Hennebert	2
URL Rewriting	8
Implementierung Generierung von Grafiken/Statistiken	8
Mailing Lists	16
Export nach RERO	8
TOTAL	43

Probleme / Lösungen

Beschreibung des Problems	Beschreibung der Lösung (wenn gefunden...)

Diverses

RERO Implementierung sehr komplex. Herr Formaz von IDIAP hat schon eine Lösung für den Export nach Marc21 auf <http://publications.idiap.ch> implementiert. Habe ihn per Mail kontaktiert. Warte noch auf Antwort.

Fragen

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen

Samuel Werner

Verantwortlicher Dozent : Jean Hennebert

Woche : 10
(2. August 2010)

Deklaration Wochenstunden

Aufgabe	Stunden
Projektverwaltung	1
Export nach RERO	16
Testen und Optimieren der Website	20
Abschlussdokumentation	4
TOTAL	41

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen

Samuel Werner

Verantwortlicher Dozent : Jean Hennebert

Woche : 11
(9. August 2010)

Deklaration Wochenstunden

Aufgabe	Stunden
Projektverwaltung	1
Abschlussdokumentation und DVD	48
TOTAL	49

Informationssystem für die Verwaltung von wissenschaftlichen Publikationen Pflichtenheft

Einleitung

Die HES-SO Sierre hat kürzlich ein Verwaltungssystem für wissenschaftliche Publikationen realisiert, was sich unter <http://publications.hevs.ch> befindet. Die Webseite basiert dabei auf dem Open Source Tool Aigaion, welches öffentlich verfügbar ist und weltweit verwendet wird. Das aktuelle System ist sehr leistungsstark für die grundlegende Verwaltung von erstellten Publikationen. Dabei ist es möglich, Publikationen sehr einfach in das System aufzunehmen, zu verändern oder zu löschen. Im Hintergrund läuft eine MySQL-Datenbank.

Die Funktionalitäten auf höherem Niveau sind jedoch stark limitiert. In der Arbeit geht es darum, innovative Erweiterungen an der bestehenden Webseite zu implementieren.

Funktionalitäten

Die Funktionalitäten können thematisch in folgende drei Punkte aufgeteilt werden:

1. Generierung von Grafiken und Rapporten

Damit die Informationen besser visualisiert werden können, sollen in einem ersten Schritt verschiedene Lösungen analysiert werden, mit denen man Grafiken im Stil von Dashboards erstellen kann. Dabei soll darauf geachtet werden, dass die neue Technologie kompatibel mit der vorhandenen Website und Aigaion ist. Es soll dann eine Lösung ausgewählt werden, mit der die Rapporte in Zusammenhang mit den Statistiken der jährlichen Publikationen generiert werden. Dies soll anhand von Institut aber auch anhand von Person erfolgen.

2. Export von Daten

Ziel ist es, verschiedene Möglichkeiten zu analysieren, um einen reibungslosen und optisch ansprechenden Export der Publikationen, den Autoren und den Informationen in eine Excel-Tabelle zu ermöglichen. Die beste Lösung soll dann implementiert werden.

3. Sichtbarkeit der Seite erhöhen

Hier sollen verschiedene Techniken analysiert und implementiert werden, die die Wahrnehmung der Seite verbessern. Im Spezifischen sind dies:

- Suchmaschinenoptimierung: Die Seite soll von Suchmaschinen besser gefunden werden. Dazu sollen eine Reihe von Methoden implementiert werden wie z.B. URL Rewriting, d.h. URLs zu Autoren werden automatisch umgeschrieben, damit sie vom Benutzer und Suchmaschinen besser gefunden werden, z.B. <http://publications.hevs.ch/index.php/authors/show/7> wird zu <http://publications.hevs.ch/henning.mueller>
- Mailing Lists: Mitglieder erhalten automatisch eine E-Mail, wenn z.B. eine neue Publikation veröffentlicht wurde.

Optionale Funktionalitäten

- Aufbereitung und Veröffentlichung der Lösungen im Sinne einer Erweiterung von Aigaion
- Automatisches Erkennen von Browser und Gerät (PC, Mobile Phone, etc.), damit die Seite entsprechend dargestellt wird.
- Implementierung von Schlagwortwolken (tag clouds), um die Kompetenzen und Themen der einzelnen Autoren zu verdeutlichen. Es gibt zwar schon eine Möglichkeit Tag clouds zu integrieren, jedoch wird dies anhand von selber definierten Stichwörtern realisiert. Das neue System soll die Schlagwortwolke automatisch anhand der veröffentlichten Arbeiten des Autors generieren.

Planung des Projekts

1. Teil: Einführung ins Projekt

- Studieren und Verstehen der jetzigen Webpage und Aigaion (Code, etc.)

2. Teil: Analyse

- Suche nach verschiedenen Tools und Möglichkeiten zur Generierung von Rapporten und Grafiken
- Auswertung von möglichen Tools zum Export von Daten
- Untersuchung der generellen Funktionsweise von Suchmaschinen und Analyse verschiedener Methoden zur Suchmaschinenoptimierung

3. Teil: Implementierung

- Implementierung: Generierung von Grafiken und Rapporten
- Implementierung: Export von Daten
- Implementierung: Tools zur Erhöhung der Sichtbarkeit der Seite

4. Teil: Unterlagen

- Planung Soll/Ist des Projekts
- Abschlusssdokumentation und CD
- Pflichtenheft

Verwaltung des Projekts

Planung Soll/Ist

Am Anfang des Projekts wird eine Planung über den gesamten Zeitraum des Projekts erstellt. Wöchentlich wird überprüft, wie der aktuelle Stand ist und wo es Probleme gibt. Die effektiv gebrauchte Zeit wird jede Woche festgehalten, damit der Unterschied zwischen Soll und Ist jederzeit ersichtlich ist.

Wöchentliche Sitzung

Jede Woche findet eine Sitzung mit dem Dozenten Jean Hennebert und/oder Henning Müller, der die Rolle des Kunden einnimmt, statt um Probleme, den aktuellen Stand und das weitere Vorgehen zu besprechen.

Samuel Werner, 21. Mai 2010

Unterschriften

Dozent:

Student:

.....
(Jean Hennebert)

.....
(Samuel Werner)